Nanofabric Power Analysis: Biosequence Alignment Case of Study

Nanofabric Power Analysis: Biosequence Alignment Case of Study

Stefano Frache, Luca Gaetano Amarù, Mariagrazia Graziano *Member IEEE* and Maurizio Zamboni
Electronics Department, Politecnico di Torino, c.so Duca degli Abruzzi 24, Torino, Italy
Email: {stefano.frache, mariagrazia.graziano, maurizio.zamboni}@polito.it

***Abstract*—The promising features of Nanoscale array structures pave the way for interesting applications like biosequence alignment, that currently can be addressed only at the price of a huge overhead in terms of area and power dissipation. Nanofabrics, once technology will be mature enough, are expected to enormously overcome these limits, and assure an evident advantage in terms of processing capabilities.**

Therefore biosequence alignment is our case of study in this work and we use the NanoASIC (NASIC) as target platform. We developed an event based simulator which works at nano-wire FET (nwFET) level to evaluate logic behavior. Here it evolved so that a detailed switching activity of simple library gates could be found in order to evaluate their power dissipation. This is devised using accurate ballistic nwFET models used to fully characterize nwFET ON and OFF characteristics and gate capacitance. From our results it is evident an underestimation of these values if quantum effects are not taken into account.

We then proposed an architectural solution to a biosequence alignment problem, based on the concurrent execution of identical processing elements (PE) instanced in an arbitrary number. Performance in terms of power, area, timing and processing capabilities were found for a single processing element as a function of several design and technology parameters. The design solution space was then explored considering an increasing number of parallel PE. The expected improvements in terms of power, area and timing with respect to solutions proposed using currently available technology have been underlined. From one to three orders of magnitude is the expected improvement in terms of processing capability (depending on the possible technological scenarios), with a power dissipation reduction from 3 to 12 times, respectively.

I. Introduction

Thanks to the integration reached by scaled technologies, parallel computation is now a reality with multiprocessors systems. Nevertheless, even though research and technology is expected to greatly improve in this field in years to come, the predicted limits of CMOS technology [1] will bound the amount of information that can be processed in parallel. Many emerging nanoscale array structures, on the contrary, show promising perspectives in the direction of massive parallel computing structures [2]. Manifold nano-structures have been devised in recent years, and there is still not a clear winner. Among the proposed solutions, many are based on nanowire arrays [3], organized in matrices [4], where active nanodevices (diodes and FETs) are created in their crosspoints. NASICs designs have been proposed in [5] [6] as an improvement over general-purpose programmable fabrics (PLAs) since, according to their proponents, they lead to denser designs with better fabric utilization and circuit cascading. These structures are basically two dimensional tiled arrays, and authors in [2] show how massively parallel architectures can be made out of these fabrics. Details about these structures are given in section II.

Due to the specific nanowires characteristics, researchers have to address many issues [6], both at device level and at architectural level. One of the points that most of the previously cited works share (except, partially, for [18]) is the lack of simulation tools to study the behavior of such circuits. This led us to develop a software tool, that we presented in [7], which enables to study, among other parameters, the impact of the high defect rates of wires and devices (nwFET) on the output error rate anf yield of NASIC circuits using a switch level modeling. Details about the simulator, together with related results, are given in section III.

We are now able to perform different kind of simulations both at circuit level and at architectural level. In the present work we used this software to perform power characterization on NASIC circuits: we first studied elemental circuits, such as XOR, AND and FA. To reach this goal we characterized the nwFET by means of an accurate model, which takes into account the effects of parasitics and interface traps on ballistic nwFET in the Ultimate Quantum Capacitance Limit [8] [9]. The reason for this model is that traditional formulas for the calculation of junction capacitance C_j that come from an approximate solution of $C = Q/V$. This approach does not take into account quantum effects, though they are extremely relevant for nanometer structures [8]. Details are in section IV.

We used the results of the characterizations to devise power consumption at architectural level. In fact, in this paper, an architecture (details in section V) has been used as a case of study for our analysis of NASIC fabrics power consumption. This architecture is capable of performing a Pairwise Sequence Alignment algorithm [11], and the choice of the specific algorithm in this context is briefly motivated herein.

Proteins, the biochemical compounds consisting of one or more polypeptides, are the building blocks of life: with their interactions, in fact, they actually define the biology of life as we know it. The information necessary to the creation of Proteins lies encoded within the genes. Researchers, then, need tools to sequence and annotate genes. Sequence alignment is a way of arranging sequences of biological interest to identify regions of similarity that may be a consequence of functional, structural, or evolutionary relationships between them.

Many techniques have been devised to solve the sequencing problem. One of these has traditionally been to develop heuristics to reduce the search space. This, of course, impacts on the execution time of sequence alignment algorithms [10]. There is a drawback too, in this approach: the quality of results is inversely proportional to the speed of execution of the heuristics [11].

In this paper we focus on one among the many exhaustive search algorithms: Smith-Waterman sequence alignment. It is implemented in a linear systolic array for general purpose Pairwise Sequence Alignment: details are in section V. There is a global community contributed database of sequences that researchers heavily rely upon, and the demand for computational power on the servers is steadily increasing. Sequence alignment against a database belongs to a well known class of problems: “embarrassingly parallel” problems, a reality in the current scenario of applied science [12]. It is therefore possible to compare the sequence of interest against each database entry in a parallel and independent fashion.

This, therefore, is the perfect domain for a massively parallel architecture such as the one used in this paper. In fact, many attempt

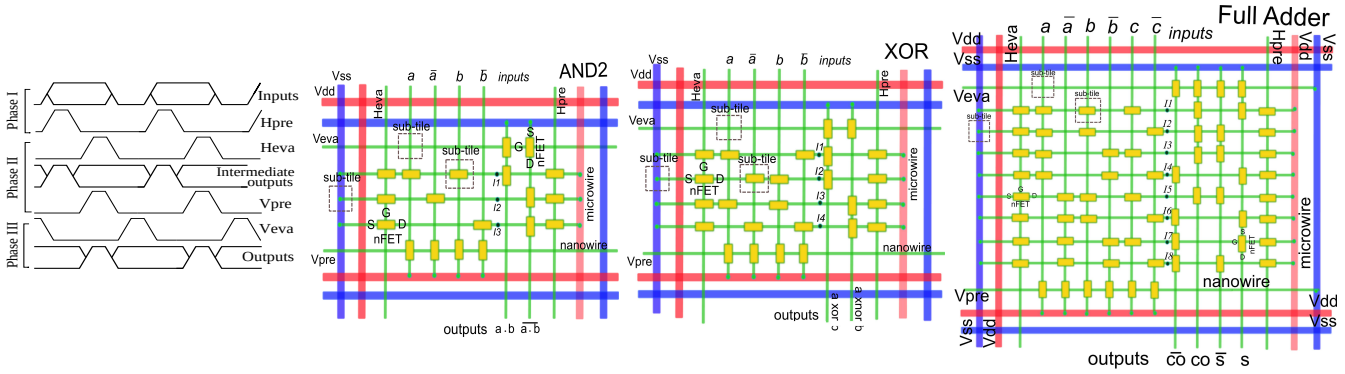

Fig. 1. NASIC tiles. Left: control signals sequence (three phases). Right: two inputs AND, XOR and Full Adder; outputs are doubled (true and false).

have been made [13] [14] to exploit parallelism to improve the execution time of these algorithms, and various hardware accelerators have been deployed too [15] [16], mostly in FPGAs. Moving from [14], we tried to scale their proposed architecture to NASIC and we have then made this architecture our case study for the power model. In section V we provide a discussion on the possible solution space and envision a future scenario for the performance that can be expected from nanofabric structures.

II. NASIC structure

According to its proponents [5], the elemental units in NASIC are the *tiles*. These are circuits for adders, multiplexers, and flip-flops. In Figure 1 the tiles implementing an AND, an XOR and a Full Adder are shown. Individual tiles can then be connected with nanowires (NW) or microwires (uW) to form a larger, multi-tile structure. Their nano-scale underpinning is based on a grid of NWs (or CNTs). The grid crossings can be programmed either as FETs, P-N type diodes, or can be disconnected, thus implementing a two-level logic architecture.

NASIC designs do not have logic planes of fixed size and wiring/routing between them, as in PLA-type designs. Furthermore, NASICs have been proposed in both static-ratioed and dynamic styles [5], with the latter that enables pipelining and overcomes the many limitations of a static design. An example of a NASIC tile for a two input XOR function ($a \oplus b$, $\overline{a \oplus b}$) is sketched in Figure 1.

The outer part of the tile is for power supply distribution purposes: micro-wires are used to carry power. Then H_{eva}, H_{pre}, V_{eva} and V_{pre} are control signals carried by nanowires from the CMOS level. These control signals are needed to propagate information across the two logic planes (AND-OR) in the structure. In this example we have chosen a NAND-NAND implementation.

Dataflow in NASICs is through a 3- or 4-phase progression and the control signals from the CMOS level coordinate these phases. With reference to Figure 1 left, we see a three-phase dynamic control scheme has been used in this case. In a cycle, horizontal precharge happens first (H_{pre} high), then horizontal wires (H_{eva} high) are evaluated and, at the same time, the precharge of the vertical wires, since V_{pre} is high too. These phases are overlapping. The cycle ends with the evaluation of the vertical wire: V_{eva} is now high.

It is worth noticing that all nanoscale computing systems have to deal with the high defect rates of nanodevices and faults introduced by manufacturing of fabrics, and so do NASICs. Faults are handled by masking them in the circuit and/or architecture design itself, implementing a multi-tiered built-in fault tolerance approach. Simulations suggest that this built-in approach would be able to achieve 25-30% yield at 10% defect rate on a fabric grid implementing a simple processor [17].

Even though our simulator allows to gather yield due to defects [7], in this paper we focus on performance evaluation only, leaving to future works the analysis of defects on the architecture considered here as a case of study.

III. Simulator

The overall aim of the simulation software we introduced in [7] is the study of complex systems based on emerging electronic nanotechnologies, with particular emphasis on architectures that can exploit their many peculiarities. More specifically, we are interested in exploring techniques proposed to solve the problems of reliability of these devices, in identifying the most suitable control schemes in dynamic systems, in studying power consumption, dimensions, performance and, as a consequence, in developing optimized architectures.

Though we aim, like in [18], to maintain the simulator general, so that it can be adapted to the evolving fabric styles proposed in literature, we underline that the key feature of our simulator is the ability to take into account technological characteristics, dynamic style, topology, and, at the same time, the possibility to efficiently analyze the behavior of complex architectures.

The simulation tools, implemented in C++, works as an *event* driven simulation engine: for a logic simulation, an *event* can be every change in value of a part of the circuit. The most outer *events* are changes in value of the input signals. Then their propagation must be supported by a medium. For instance, an event on one of the inputs can be propagated along the wires (both micro and nano wires). Until there is support for the propagation, the event will flow. The simulation ends when no event need to be handled anymore. Of course, along the wires many things can happen. There can be a nwFET gate, so the propagation of the event may change the conductivity of the channel (in Source-Drain direction). It could also enter the channel, and the propagation could end there or not, according to the value of the gate.

In this way we can handle internal data signals as well as control signals and even power supply along microwires. Moreover, the control scheme is not embedded in any way inside the software and this allows for maximum flexibility in design choices.

At the end of the simulation all significant waveforms and statistics about the main signals are available: in table I values of the input, control and output signals for AND and XOR gate, as well as waveforms in Figure 2 for the Full Adder (FA) are shown.

TABLE I
INPUTS/OUTPUTS FOR AND AND XOR TILES AFTER SIMULATION.

Φ		Inputs							AND Outputs		XOR Outputs	
		Heva	a	$\overline{a}$	b	$\overline{b}$	Hpre	Veva	Vpre	$a \cdot b$	$\overline{a} \cdot \overline{b}$	$a \oplus b$	$\overline{a \oplus b}$
1	I	0	**1**	0	**1**	0	1	0	0	1	1	1	1
	II	1	1	0	1	0	0	0	1	1	1	1	1
	III	0	1	0	1	0	0	1	0	**1**	**0**	**0**	**1**
2	I	0	**1**	0	**0**	1	1	0	0	0	1	1	1
	II	1	1	0	0	1	0	0	1	1	1	1	1
	III	0	1	0	0	1	0	1	0	**0**	**0**	**1**	**0**
3	I	0	**0**	1	**1**	0	1	0	0	1	1	1	1
	II	1	0	1	1	0	0	0	1	1	1	1	1
	III	0	0	1	1	0	0	1	0	**0**	**0**	**0**	**1**
4	I	0	**1**	0	**1**	0	1	0	1	1	1	1	
	II	1	1	0	1	0	0	0	1	1	1	1	1
	III	0	1	0	1	0	0	1	0	**0**	**1**	**0**	**0**

In order to evaluate power, from a statistical point of view, we must collect switching activity information on an appropriately large data set of random input values to correctly characterize circuit behavior.

The simulation software we presented in [7] now features an expanded function set which, among other improvements, comprises detailed analysis of the switching activity of the circuit under test. This enables to thoroughly characterize NASIC designs with respect to power consumption.

It is necessary to gather information about switching activity occurring in input nodes, internal nodes (I_i in Figure 1) as well as output ones. We chose those internal nodes because, by recording their switching activity, you can calculate power consumption, since you know exactly how many C_g you are charging and discharging (i.e. the vertical wires of the outputs).

So, first of all, the circuit being simulated is fed with input sequences that correctly follow the chosen control scheme (three or four phases), and still allow for randomness in the input values, provided they are consistent with their complementary nature. The input stimuli generator, which is part of the tool, can perform this task and, besides the input values, gives as output the activity of each selected node. The size of the input vector can be chosen in order to achieve statistical significance.

Since the propagation of information inside the structure is handled on an *event* basis, information must be collected about the switching events in appropriate points on the fabric (i.e. in internal nodes I_i). We can "probe" the circuit by choosing where to record switching events in the horizontal nanowires of the circuit under test. In this case, the nodes in-between the horizontal and vertical logic planes were chosen. Also the outputs of the circuit were collected and analyzed to get their switching activity.

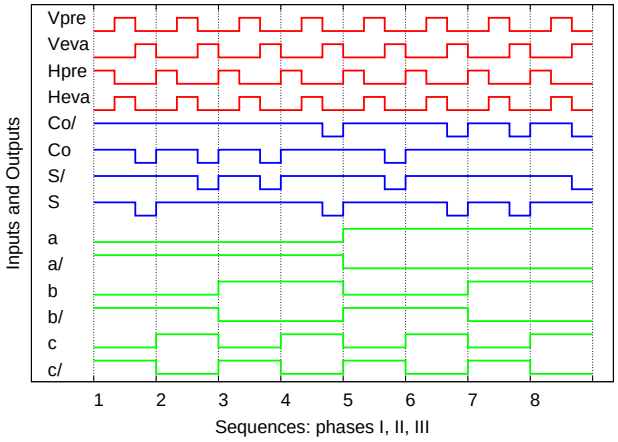

Fig. 2. Full Adder waveforms after an exhaustive simulation.

In this way we are not estimating the switching activity of the nanowires: we are actually recording every change in their charge.

Table II (left) shows switching activity results for the three gates used here as a reference: we used 1000 trials for AND and XOR, and 4000 for the Full Adder. The number of nwFET gates driven by each wire is also shown. These data have been used to perform the calculations, as detailed in what follows.

IV. NWFET CHARACTERIZATION AND POWER ESTIMATION

In NASIC tiles, as in most clocked electronic logic systems, power consumption is due to two main sources: dynamic charging/discharging of capacitances and leakage currents. In order to accurately estimate these sources, a complete physical characterization of a NASIC tile is needed. The main component in this kind of structure is certainly the nwFET which connects input and output planes or vice versa.

Previous works [5] use results from [19] to determine the on-resistance. In fact hypothesizing a nwFET with 4 nm width and 5 nm gate length, they calculate $R_{on} = \rho_{Si} \cdot \frac{5nm}{\pi 2nm^2}$ using the second Ohm's law. The resulting on-resistance is $R_{on} \approx 4k\Omega$. This modeling approach based on geometrical dimensions holds as a first approximation but in a nanometer device as nwFET, it is very probable to underestimate the real values. In fact, equations implemented in [9] and in [8] take into account quantum and ballistic transport effects and lead to higher R_{on} values.

Previous works assume a value of $10G\Omega$ for off-resistance with reference to the work in [20]. Also junction capacitance is calculated using the geometrical formulas proposed in [20]: under the aforementioned conditions, a value of $0.652aF$ was extracted. The same reasoning is valid for gate capacitance estimation: a geometrical method, as the one in [20], has been demonstrated to be no accurate enough for a nanometer device: using the method in [8] and [9] leads to an almost tripled value.

For what concern the dynamic performance, previous works estimate delay using a lumped RC model. For WISP-0 implementation, the work in [17] estimated an operating frequency of $93GHz$ for a three phase control scheme. Dynamic power was calculated as:

$$P_{dyn} = \sum_{pre,eva}(C_{L1} + N * C_{L2}) * V_{dd}^2 * f$$

where C_{L1} is the capacitance on control NWs, C_{L2} is the capacitance on a datapath NWs and N is the number of datapath NWs switching simultaneously, while the sum is executed considering all the precharge and evaluation stages. The leakage power is estimated in the order of tens of nanowatt for WISP-0 implementation.

In our work we used these models as a reference point, but we improved the accuracy on the nwFET parameters by studying the most important features that are affected by nanoscale related phenomena. We included more accurate models and we analyzed their sensitivity to design and technological parameters. Finally, we selected typical values to be used for the power evaluation. Power analysis is then based on nanoscale-level FET models and on accurate, simulation-based switching activity data.

A. nwFET simulation

nwFETs are widely studied in literature: accuracy level in physical modeling of these objects is sharpening in time. In fact, improved models are continuously developed, in which quantum and secondary effects are taken into account.

TABLE II

LEFT TABLE: SWITCHING ACTIVITIE (SA) AND LOAD FOR EACH SIGNAL OF AND2, XOR AND FA. INPUTS ARE RANDOMLY VARIED ON A STATISTICALLY RELEVANT NUMBER OF TRIALS. RIGHT TABLE: GATES CHARACTERIZATION; DYNAMIC POWER IS PARAMETRIC ON FREQUENCY AND FET CHANNEL LENGTH, AREA ON SUB-TILE PITCH $Pi[nm]$, TIMING ON FET τ, E.G. ON FET CHANNEL LENGTH ($\tau = 8.57fs/nm$).

SWITCHING ACTIVITY AND LOAD

AND2 1000 Trials			XOR 1000 Trials			Full Adder 4000 Trials					
Signal	Load	SA	Signal	Load	SA	Signal	Load	SA	Signal	Load	SA
Heva	3Cg	0.667	Heva	4Cg	0.667	Heva	8Cg	0.667	I1	2Cg	0.087
Hpre	3Cg	0.667	Hpre	4Cg	0.667	Hpre	8Cg	0.667	I2	2Cg	0.105
Veva	2Cg	0.667	Veva	2Cg	0.667	Veva	4Cg	0.667	I3	2Cg	0.076
Vpre	4Cg	0.667	Vpre	4Cg	0.667	Vpre	6Cg	0.667	I4	2Cg	0.112
a	Cg	0.505	a	2Cg	0.505	a	4Cg	0.505	I5	2Cg	0.083
$\overline{a}$	Cg	0.505	$\overline{a}$	2Cg	0.505	$\overline{a}$	4Cg	0.505	I6	2Cg	0.103
b	Cg	0.498	b	2Cg	0.498	b	4Cg	0.494	I7	2Cg	0.082
$\overline{b}$	Cg	0.498	$\overline{b}$	2Cg	0.498	$\overline{b}$	4Cg	0.494	I8	2Cg	0.118
I1	Cg	0.163	I1	Cg	0.163	c	4Cg	0.492	$\overline{Co}$	4Cg	0.325
I2	Cg	0.341	I2	Cg	0.160	$\overline{c}$	4Cg	0.492	Co	4Cg	0.342
I3	Cg	0.344	I3	Cg	0.163				$\overline{Sum}$	4Cg	0.336
$a \cdot b$	Cg	0.503	I4	Cg	0.182				Sum	4Cg	0.331
$\overline{a} \cdot \overline{b}$	Cg	0.163	$a \oplus b$	Cg	0.323						
			$\overline{a \oplus b}$	Cg	0.323						

CHARACTERIZATION

Value	AND2	AND3	EXOR2	FA
Dynamic Power P^d $[nW/(nmGHz)]$	2.82	3.76	3.59	8.87
Static Power P^s $[nW]$	0.86	1.05	0.92	1.45
Area A $[nm^2]$	40× Pi	60× Pi	48× Pi	120× Pi
Timing $[fs]$	3×τ	4×τ	4×τ	8×τ

A reliable tool to simulate nwFETs, already available in [9], is $FETToy$ based on a set of scripts calculating the ballistic I-V characteristics for different FET structures, including nwFETs. $FETToy$ provides good estimations for different parameters such as I_{ds}, C_g, μ_e etc.; it only requires as input geometrical dimensions, material properties, supply conditions and operating temperature. Considering how fast is the technology pace, some of the original $FETToy$ models have been surpassed by several more recent ones. Consequently we updated, where necessary, $FETToy$ with models in [8] in order to characterize nwFETs with maximum possible accuracy.

For what concerns I_{ds} current, the model presented by [8] does not lead to considerable variations with respect to the one implemented in $FETToy$. On the other hand, gate quantum capacitance values estimated by $FETToy$ were quite optimistic: the model proposed and validated by [8] suggests an almost doubled value. Moreover, in $FETToy$ there was no τ model, which on the contrary is present in [8]. Finally, we renewed the existing scripts using an Octave implementation with $C_{g-quantum}$ and τ equations from [8].

nwFETs gate capacitance has been traditionally [20] expressed according to the following formula:

$$C_j = \varepsilon \cdot 2\pi \cdot d \left(\ln \left(1 + \frac{2t_{junc}}{d} \left(1 + \sqrt{1 + \frac{d}{t_{junc}}} \right) \right) \right)^{-1}$$

where t is the width of the shell around the conductor ($t_{sh} = t_{junc}$ in nanowires) and d is the nano/micro wire diameter. As we said before, this geometrical approach does not take into account quantum effect that, at this scale, are extremely important, and is optimistic, as we are going to see. The method we chose to overcome these limitations [8] estimates the gate capacitance as follows:

$$C_g(V_g) = \sum_{\Omega} C_g(i) \qquad \text{and} \qquad C_{gi}(V_g) = \frac{\delta Q_{gi}(V_g)}{\delta V_g}$$

where Ω is the set of all possible sub-bands, Q_{gi} is the total mobile charge contribution from the ith sub-band. This equation has been rewritten in terms of the Fermi integral, as follows:

$$C_g(V_g) = \frac{q}{\pi\hbar} (2m^* k_B T)^{0.5} \sum_{\Omega} \frac{\delta}{\delta V_g} \left[F_{-\frac{1}{2}} \left(\frac{\mu_s - E_i^0}{k_B T} \right) \right]$$

Then this method has been implemented in FETToy tool. For the computation of the expression with the derivative of the Fermi integral, a numerical method from [21] has been adopted. The correctness of the results have been assessed by contrast and comparison with a FORTRAN implemented quadrature method from [22]. In Figure 3 (top left) we see that the results, for W=1nm, t=1.5nm, T=300K, with the more accurate method are less optimistic, and the overall trend of the curves is quite similar. Again from [8] we implemented in FETToy tool the accurate τ model as follows:

$$\tau = (2m^*)^{-0.5} (k_B T)^{-1.5} \left(\frac{e^{qV_d/k_B T}}{e^{qV_d/k_B T} - 1} \right) \frac{\sum_{\Omega} \left[F_{-\frac{1}{2}} \left(\frac{\mu_s - E_i^0}{k_B T} \right) \right]}{\sum_{\Omega} \left[F_0 \left(\frac{\mu_s - E_i^0}{k_B T} \right) \right]}$$

B. Sensitivity analysis

We identified three elements as fundamentals to characterize a nwFET: width, thickness and temperature. By tuning these parameters in their operating range, in all possible combinations, we performed the sensitivity analysis of gate quantum capacitance, on-current, τ, R_{on} and R_{off}. It is worth pointing out that, as delay metric, in the previous equation we chose $\tau = \frac{\int CdV}{I}$ instead of $\tau = \frac{CV}{I}$: in Ultimate Quantum Capacitance Limit (UQCL) $\tau = \frac{\int CdV}{I}$ metric gives more accurate results.

We simulated a nwFET with the following characteristics:

- Silicon nwFET
- Electron transverse mass : $0.19m_0$.
- Width varying from 1 to 15 nm.
- Thickness varying from 0.5 to 2 nm.
- Gate insulator dielectric constant: 3.9.
- Temperature varying from 250 to 400 K.
- Supply voltage: 1 V.

In the following paragraphs capacitance, on-current, τ, R_{on} and R_{off} sensitivity results are presented.

1) Capacitance: Gate quantum capacitance has proven not to be very sensible to thickness and width variations, due to UQCL regime. Small fluctuations are present as electrons occupy different sub-bands with different geometrical dimensions. It is worth to emphasize that UQCL regime will not be valid anymore for width values greater

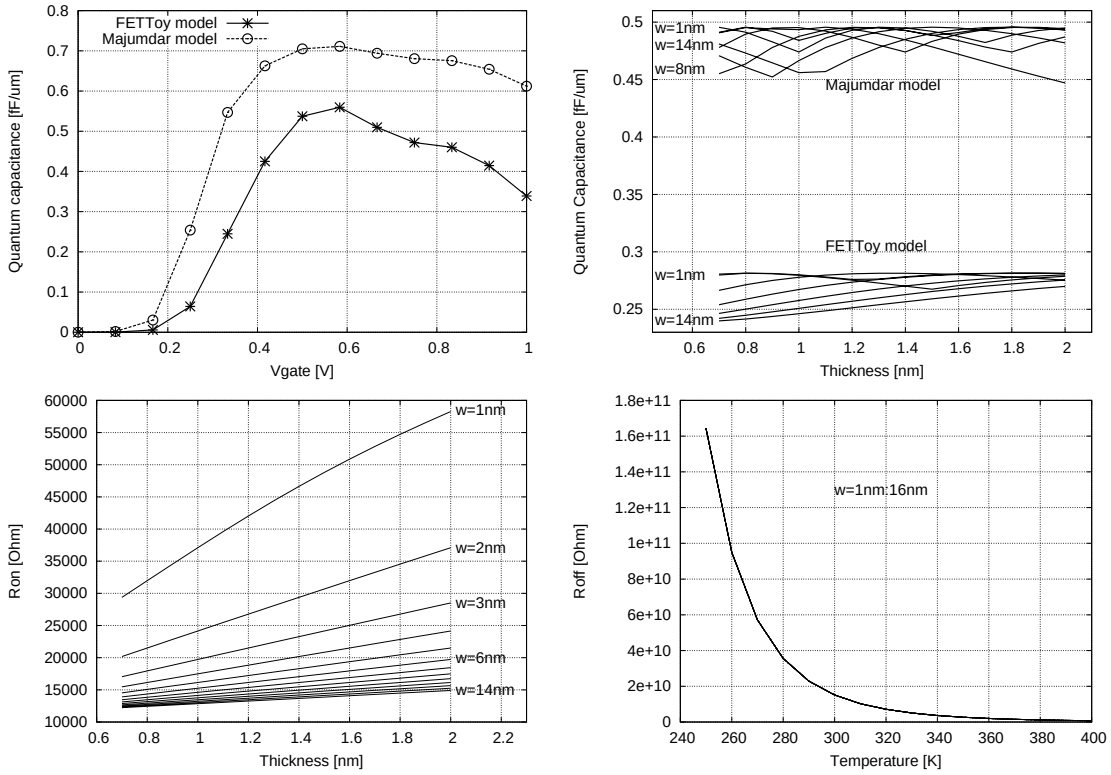

Fig. 3. nwFET characterization. Top left: quantum capacitance using FETToy and Majumdar models. Top right: sensitivity of quantum capacitance on transistor thickness and width. Bottom left: sensitivity of R_{on} on FET width and thickness. Bottom right: sensitivity of R_{off} on temperature and FET width.

than 15 nm: in that case classical regime will be more appropriate. The method from [8] shows a behavior similar to FETToy's original model (Figure 3 top-right), with a doubled value: about $0.5fF/\mu m$ averaged with respect to gate voltage, which is coherent with results obtained by [8].

As long as temperature is concerned, instead, both models present a linear dependence on temperature variations which can be approximated to $\Delta C/\Delta T = -0.0012\frac{fF}{\mu m \Delta K}$.

2) On-current, τ, R_{on}: As τ and R_{on} are proportional to I_{on}^{-1}: we will analyze them in conjunction with I_{on}. I_{on} showed high sensitivity to both width and thickness variations: Figure 3 (bottom-left) shows on-resistance simulation results at 300K. Best performance points, for both τ and I_{on}, are in the minimum R_{on} condition: maximum width and minimum thickness. The above parameters are not very sensitive to temperature variations, instead: this result is supported by the work in [23]. In fact, temperature variation implies a shape variation in I-V characteristic but should not appreciably modify the maximum current point [23].

3) R_{off}: Off-resistance shows an almost constant value of $15.12G\Omega$; actually little variations exist, with respect to thickness and width. On the other hand, temperature variations heavily affect the off-resistance: Figure 3 (bottom-right) shows simulation results.

4) Typical values: Typical silicon nwFET dimensions in literature are 4 nm width and 1 nm thickness. Operating temperature is assumed 300 K and supply voltage equal to 1 V. Under these conditions the nwFET presents:

C_{gate}: 0.49 $fF/\mu m$	I_{on}: 51.16 μA
R_{on}: 17.49 $k\Omega$	R_{off}: 15.12 $G\Omega$
τ: 8.57 $ps/\mu m$	

C. Power and timing characterization

On the basis of previous models, dynamic and static power, and timing are evaluated according to the equations described herein. Results for basic gates found using these equations and the above mentioned switching activity data are reported in table II.

1) Dynamic power: Regarding dynamic power we use:

$$P_{dyn} = \sum_{all-NWs}(A \cdot \tfrac{1}{2} C_{wire} V^2 \cdot f)$$

where A is the nanowire activity factor and

$$C_{wire} = N_{nwFET} C_{gate}$$

N_{nwFET} is the number of crossed functions for each NW. This formulation is supposed to be accurate: it takes into account the activity of each nanowire and so a switching capacitance closer to the real value. Using a pitch larger than 10 nm, as well as using some shield material between nanowires, the parallel nanowire coupling capacitance are expected to be negligible.

2) Timing evaluation: We estimated delays using a lumped RC model. In the characterization of basic NASIC tiles, we obtained that the worst delay comes from the Full Adder. In fact, FA horizontal precharge and evaluation phases require charging up to 8 gate capacitances. Consequently, for a 5 nm gate length nwFET, we have a worst case delay of $342.80fs$. The maximum theoretical operating frequency with this methodology, assuming a 33% duty cycle, will be approximately $0.97THz$.

3) Static power: In order to estimate static power for a NASIC tile, informations about output probability for each nanowire should be available. This task can be addressed by high level logic simulation. High level logic simulation should also provide informations about the number of "off" nwFET at the same time when nanowire output is logic '1'. Finally, static power can be calculated as follows:

$$P_{static} = \sum_{all-NWs}(\tfrac{P_{NWout}(0) \cdot V^2}{R_{off}}) + (\tfrac{P_{NWout}(1) \cdot V^2}{N_{off} \cdot R_{off}})$$

When logic simulation is not available, the static power consumption can be overestimated as $P_{static} = N_{nws} \cdot \frac{V^2}{R_{off}}$. We chose to overestimate in this way static power to give a "worst case" idea about the order of magnitude involved.

V. Biosequence alignment architecture: a NASIC implementation

Figure 4 (top) presents a linear systolic array implementation for Pairwise Sequence Alignment. The array consists of a pipeline of basic Processing Elements (PE): each of them holds query sequence residues, whereas the subject sequence is taken from the database and shifted systolically through the array. Each PE holds one or more residue of the query sequence and performs one elementary calculation in one clock cycle. The full alignment of two sequences of lengths K and M is achieved in $M+K-1$ cycles. The following subsection details the algorithm implemented in each PE.

A. Processing Elements (PE) architecture

As described in [14] each processing element (PE), when *active*, should perform the Smith-Waterman algorithm shown herein:

```
F(i,j)=Max{F(i-1,j-1)+s(xi,yj),
           F(i,j-1)-d,F(i-1,j)-d,0}
Max(i,j)=Max{Max(i-1,j),F(i,j-1),Max(i,j-1)}}
```

Consequently the following operations are required: 3 algebraic additions, a 4 maximum search (MAX4) and a 3 maximum search (MAX3). We now discuss a possible hardware implementation of the former structures using as a library the gates previously characterized. If N is the bit width of inputs and outputs, the algebraic addition will be implemented by a ripple-carry adder (RCA) having a cost of N full adders.

For a MAX3, 3 algebraic additions are required. Said x_p with $p = 0, 1, 2$ the inputs, the algebraic additions perform $x_p - x_q$ with $p = 0, 1, 2$ and $q = p + 1$. Said s_{pq} the sign of $x_p - x_q$, the following property holds: $\overline{s_{pq}} = s_{qp}$. Now it is possible to know if x_p is the maximum by: $\bigwedge_{q=0,q\neq p}^{2} s_{q,p}$ where $\bigwedge$ stands for the AND operation. If the AND result is logic '1' then x_p is the maximum. Obviously only one maximum can simultaneously exist. With the one-hot information of the maximum location it is possible to AND, bitwise, each output with the corresponding maximum bit. In this way 3 set of N bits will be available: two set are formed by all logic '0' and only one contains the maximum. Using these set, bitwise, as inputs of an OR port it is possible to obtain the value of the maximum.

We chose to implement this unit only with AND2, AND3 and FA NASIC tiles. The corresponding cost, in terms of components, will be $3N \cdot FA + 3 \cdot AND2 + 3N \cdot AND2 + N \cdot AND3$. The AND-OR structure has been implemented by a NAND-NAND configuration. The worst case latency path is $N \cdot t_{\text{FA}} + 2t_{\text{AND2}} + t_{\text{AND3}}$.
A sketch of this unit is proposed in Figure 4.

The MAX4 unit should perform a particular 4 maximum search: the inputs are 3 numbers and logic '0'. Therefore this problem reduces to a 3 maximum search with a check to the 3 inputs signs: if all the signs are 1 the output should be N logic '0's. This task can be easily accomplished by a 3 input NAND port for the signs and a final AND port for the output. A sketch of this unit is proposed in Figure 4. The component cost of this unit is $AND3 + N \cdot AND2 + max3_{unit}$ and the latency path is $t_{\text{MAX3}} + t_{\text{AND2}}$.

In Figure 4 the elements presented above are merged to perform the Smith-Waterman algorithm for one PE. To sum up, a PE needs:

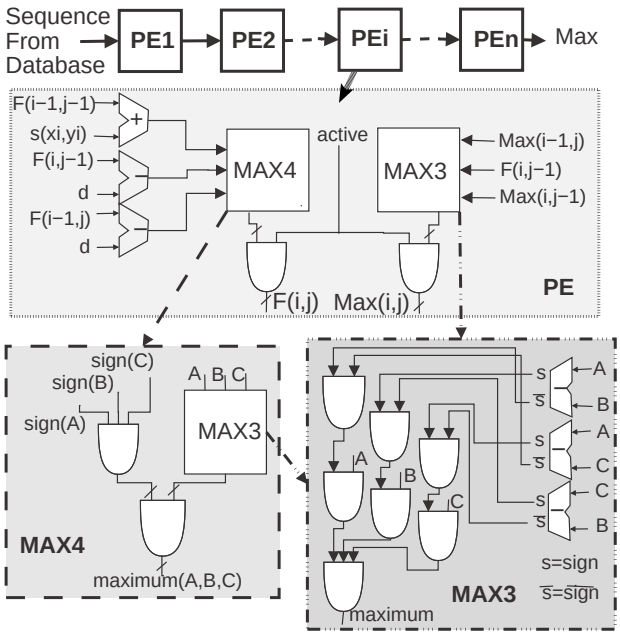

Fig. 4. General architecture and details of a possible implementation of each PE according to Smith-Waterman algorithm.

$max3 + max4 + 2N \cdot AND2 + 3N \cdot FAs$. Expanding the total PE complexity is in equation (1)

$$PE = 9N \cdot FAs + (9N + 6) \cdot AND2 + (2N + 1) \cdot AND3 \quad (1)$$

$$t_{\text{PE}}^{lat} = t_{\text{ADD}} + t_{\text{MAX4}} + t_{\text{AND2}} = 2N \cdot t_{\text{FA}} + 4 \cdot t_{\text{AND2}} + t_{\text{AND3}} \quad (2)$$

while in equation (2) the total worst case latency path is reported.

B. Solution space exploration and performance evaluation

Starting from the NASIC tile characterization shown in section IV we evaluated area, power and frequency for a single PE and, afterwards, for a parallel implementation based on an arbitrary large number of PE.

1) Single PE performance estimation: The evaluation relies on several parameters. Some of them have been fixed (e.g. the typical conditions mentioned in section IV.B), while others were used as parameters in order to explore the solution scenarios. The parameters used are: L_g as the nwFET channel length, N as the number of I/O bits, K_{ov}, an overhead factor introduced to consider a relaxation with respect to the maximum frequency value. Finally N_{fet} is the number of nwFET driven by the Heva signal in the worst case, which for the gates we are using are 8 in the Full Adder gate. Equations from (3) to (11) model the PE characteristics.

$$\tau = 8.57 \cdot L_g \quad (3)$$

$$P_i = 2.5 \cdot L_g \quad (4)$$

$$T_{cp} = N_{phases} \cdot N_{fet} N \cdot \tau \quad (5)$$

$$T = T_{cp} \cdot K_{ov} \quad (6)$$

$$T_{\text{PE}} = 2 \cdot N \cdot T + 5 \cdot T \quad (7)$$

$$F_{\text{PE}} = 1/T_{\text{PE}} \quad (8)$$

$$A_{\text{PE}} = P_i \cdot (9Na_{\text{FA}} + (9N + 6)a_{\text{A2}} + (2N + 1)a_{\text{A3}}) \quad (9)$$

$$P_{\text{PE}}^d = 9NP_{\text{FA}}^d + (9N + 6)P_{\text{A2}}^d + (2N + 1)P_{\text{A3}}^d \quad (10)$$

$$P_{\text{PE}}^s = (9Na_{\text{FA}}P_{\text{FA}}^s + (9N + 6)a_{\text{A2}}P_{\text{A2}}^s + (2N + 1)a_{\text{A3}})P_{\text{A3}}^s \quad (11)$$

TABLE III
PE PERFORMANCE IN THREE CASES: CASE1 IDEAL, CASE2 REASONABLE, CASE3 CONSERVATIVE. VALUES ARE FOR $N = 16$ BITS.

	Case1 $L_g = 1nm$ $K_{ov} = 1$ $uW_{ov} = 40\%$	**Case2** $L_g = 5nm$ $K_{ov} = 10$ $uW_{ov} = 30\%$	**Case3** $L_g = 10nm$ $K_{ov} = 10$ $uW_{ov} = 10\%$
t_{FA}	$0.216ps$	$10.848ps$	$21.695ps$
Latency	$8.027ps$	$401.36ps$	$802.72ps$
Area			
w uW_{ov}	$88410nm^2$	$410475nm^2$	$694650nm^2$
w/o uW_{ov}	$63150nm^2$	$315750nm^2$	$631500nm^2$
$Power_{tot}$	$8.4094mW$	$0.84128mW$	$0.84128mW$

In details, equation (3) describes the transistor time constant in typical conditions, while (4) represents the nano-tile pitch. The critical path T_{cp} in equation (5) is evaluated on the basis of the worst case load for a signal in terms of gate capacitance. In our solution the worst case is the Full Adder with $N_{fet} = 8$; this constrains one of the N_{phases} (in our case 3) but, for the sake of regularity of control signals delivery, we considered the same duration for the three phases. We then envision the working period as the overall delay of a single tile, even though internally the three phases are sequenced. Furthermore, as a N bit RCA is used, the total delay for the sum is multiplied by N, if the RCA is not pipelined. The stages driven as output by the RCA can be pipelined, and thus NASIC registers should be provided [24]. In this situation, the worst case delay remains the RCA one. If, on the contrary, no pipeline is adopted, the working period would be given by the latency path in equation (2). The two scenarios are herein referred to as *fully pipelined* and *not-fully-pipelined*.
The realistic working period is estimated as in equation (6), taking into account an overhead factor due to the connection between wires and the interfaces to CMOS stages. The PE latency (or period in the non-pipelined version) is in equation (7), derived by equation (2). The PE area in equation (9) is estimated starting from equation (1) as a function of P_i and using factors $a_{A2} = 40$, $a_{A3} = 60$ and $a_{FA} = 120$ defined in table II (right). Dynamic power P^d_{PE} and static power P^s_{PE} are evaluated in equation (10) and (11), respectively, using the PE composition and area and power data evaluated for the single gates (table II right).

Table III reports results for a single PE obtained using the previously defined equations. Since we obtained several parameters, these data are shown taking into account three possible scenarios: an extremely ideal Case1, a Case2 that can be considered reasonable, and a more conservative and pessimistic Case3. The value of the parameters chosen for the three cases are in the table. The word parallelism we chose is $N = 16$ bit, as suggested in [14].

A further parameter, up to now not yet introduced, is an overhead factor due to the microwires surrounding each tile. Instead of defining precise geometrical values that, due to the approximation of the current design methodology, could be not reliable, we used an overhead factor uW, simply to model the impact of microwires on the total area. Clearly, the bigger the L_g value, the bigger is the total area, and thus the smaller is the expected impact of microwires. To be more precise, both the area values are reported in table, with and without taking into account this overhead.

2) Parallel architecture performance estimation: Starting from the data found for a single PE we obtained an evaluation of a complete architecture comprising a massive number of PE instances that can process the Smith-Waterman algorithm in parallel. Clearly, as shown below, the total area and power are a linear function of the number

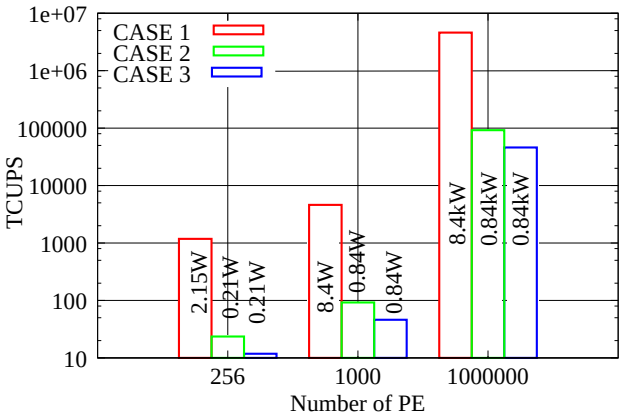

Fig. 5. Peak Cell updates Per Second (CUPS) and total power consumption for an increasing number of PE. Cases 1, 2 and 3 correspond to the different scenarios defined in table III.

of PE N_{PE}. The overall data related to frequency is here defined as Peak Cell Updates Per Second ($CUPS$).

$$A_{TOT} = A_{PE} \cdot N_{PE} \qquad CUPS = F_{PE} \cdot N_{PE}$$
$$P^d_{TOT} = P^d_{PE} \cdot N_{PE} \qquad P^s_{TOT} = P^s_{PE} \cdot N_{PE}$$

Again, the three aforementioned cases (ideal, realistic, conservative) are used as reference points. Figure 5 shows the figures we found for three levels of parallelism. The smaller N_{PE} is the reference point based on the FPGA implementation in [14]. The others are higher values, as $1 \cdot 10^3$ compatible with an ASIC implementation, and as $1{\cdot}10^6$ compatible only with a NASIC implementation, if thought of as a single device. For every case and number of PE, CUPS estimation is reported (TeraCUPS in this case) in order to suggest the processing capability of the system. Furthermore, total power dissipation values are superposed to the histogram.

As a reference, we compare these results with the work in [14], where the operating frequency of a single PE implemented in a FPGA is 40MHz, and the obtained performance is 10GCUPS. If we consider our pessimistic case (Case3) with 256 PE we get 12TCUPS, in the *fully-pipelined* version where the expected ideal operating frequency is around 46GHz. By using the PE latency as working clock period in a *not-fully-pipelined* version, the frequency decreases to 1.2GHz resulting in 318GCUPS, which is over one order of magnitude more than the result in [14]. At the same time, this demonstrates that the values we obtain are comparable to a realistic case, and that more than one order of magnitude improvement can be obtained if a nanoarray solution becomes feasible in the near future. Clearly, if the number of PE is increased (for example to 1M) the CUPS amount notably increases to 46000 TCUPS for the *fully-pipelined* version and to 1200 TCUPS in the *not-fully-pipelined* version (not reported in histogram for the sake of brevity). This confirms how this technology can greatly improve the computation capability when addressing a problem which relies on massive parallelism.

The power dissipations cannot be compared, as no data are reported in literature. Anyway, an estimate can be done by considering the power dissipation of a CMOS implementation of the PE based on the same basic gates (AND2, AND3 and FA, 90nm technology) and architecture. With a frequency of 40MHz we obtained a dynamic power of 2.56mW for a single CMOS based PE. Considering our Case3, we obtain 0.84mW as total power for the single PE at 1.2 GHz (table III).
The total power dissipation when several PE are considered is 0.21W,

0.84W and 0.84kW (Case3) with the increasing number of PE as shown in the histogram at 1.2 GHz in a *fully-pipelined* version. Even though this last value is not small, it should be compared to the corresponding dissipation using CMOS: 2.5kW with similar parameters and without taking into account the obvious overhead of a real-life system. This confirms the improvements that can be achieved, in terms of power too, by adopting this type of nanotechnology.

VI. Conclusion

We focused on a NASIC structure as a working platform to develop our simulation and design methodology, and to explore the potentiality of nanofabrics in the solution of a real-life problem: "embarassing parallelism".

Our simulator, adapted to the NASIC structure, allowed to simulate the behavior of simple gates such as AND, XOR and FA, and to evaluate their switching activity using a statistically significant number of input trials.

We characterized these gates using an improved nwFET model which includes nanoscale fenomena for gate capacitance evaluation. Power and timing values were then found, based on these models, using some significant geometrical and technological parameters as variables.

We proposed an architectural solution for a processing element to be used for a biosequence alignement case of study: an intrinsically parallel structure. Power, area and timing for this structure were evaluated as a function of the abovementioned parameters, and the solution space was explored for and increasing number of processing elements adopted in the architecture. Results showed a greatly improved performance over the same solution implemented using current technology, both in term of timing and of power. The information throughput was higly improved, with an important reduction in terms of power and area.

We thus demonstrated how this type of fabric is worth further investigations, given the expected promising performance we calculated.

References

[1] International Technology Roadmap of Semiconductor, Ed. 2009

[2] P. Narayanan et al., *Image Processing Architecture for Semiconductor Nanowire Fabrics*, in IEEE Nanotechnology conference (NANO 2008)".

[3] W. Lu et al., *Semiconductor nanowires*, in "J. Phys. D: Applied Physics", n. 39, pp. 387–406, Oct. 2006.

[4] Y. Luo et al., *Two-Dimensional Molecular Electronics Circuits*, in "ChemPhysChem", vol. 3, no. 6, pp. 519-525.

[5] C.A. Moritz et al., *Latching on the wire and pipelining in nanoscale designs*, in "3rd Non-Silicon Comput. Workshop (NSC-3)", Munich, Germany, 2004.

[6] P. Narayanan et al., *Manufacturing Pathway and Associated Challenges for Nanoscale Computational Systems*, in "9th IEEE Nanotechnology conference (NANO 2009)", July 2009.

[7] S. Frache, M. Graziano and M. Zamboni *A Flexible Simulation Methodology and Tool for Nanoarray-based Architectures*, IEEE , IEEE International Conference on Computer Design, pp. 60–67, Amsterdam 3-6 October, 2010.

[8] K. Majumdar, N.Bhat, P.Majhi and R.Jammy *Effects of Parasitics and Interface Traps on Ballistic Nanowire FET in the Ultimate Quantum Capacitance Limit*, IEEE Transaction on electron device, vol. 57, no. 9, September 2010.

[9] *FETToy 2.0 Source Code Download*, http://nanohub.org/resources/107, 2005.

[10] S.F. Altschul, W. Gish, W. Miller, E.W. Myers and D.J. Lipmanl *Basic Local Alignment Search Tool* in "Journal of Molecular Biology, n. 215, pp. 403–410, May 15, 1990.

[11] W.R. Pearson *Comparison of methods for searching protein sequence databases*, in "Protein Science", n. 4, pp. 1145–1160, 1995.

[12] J. Brodkin *10,000-core Linux supercomputer built in Amazon cloud*, Network World, April 6, 2011.

[13] V.-H. Nguyen, A. Cornu and D. Lavenier *Implementing protein seed-based comparison algorithm on the SGI RASC-100 platform*, IPDPS 2009, IEEE International Symposium on Parallel & Distributed Processing, pp. 1–7, Rome 23-29 May, 2009.

[14] K. Benkrid, Y. Liu and A. Benkrid *High Performance Biosequence Database Scanning using FPGAs*, ICASSP 2007, IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 361–364, June 4, 2007.

[15] D. Lavenier; L. Xinchun and G. Georges *Seed-based genomic sequence comparison using a FPGA/FLASH accelerator*, FPT 2006, IEEE International Conference on Field Programmable Technology, pp.41–48, Bangkok Dec, 2006.

[16] F. Xia, Y. Dou and J. Xu *FPGA-Based Accelerators for BLAST Families with Multi-Seeds Detection and Parallel Extension*, ICBBE 2008, The 2nd International Conference on Bioinformatics and Biomedical Engineering, pp. 58–62, Shanghai 16-18 May, 2008.

[17] C.A. Moritz, T. Wang, P. Narayanan, M. Leuchtenburg, Y. Guo, C. Dezan and M. Bennaser *Fault tolerant Nanoscale Processors on Semiconductor Nanowire Grids*, IEEE transactions on circuit and systems, vol.54, no.11, November 2007.

[18] C. Dezan et al., *Towards a framework for designing applications onto hybrid nano/CMOS fabrics*, Microelectronics.

[19] Y. Wu, J. Xiang, C. Yang, W. Lu and C.M. Lieber*Single crystal Metallic Nanowire and Metal/Semiconductor Nanowire Heterostructures*, Nature, vol.430, pp.61-65, 2004

[20] A. DeHon *Nanowire Based Programmable Architectures*, ACM journal on Emerging Technologies in Computing Systems, July 2005.

[21] R. Kim, M. Lundstrom *Notes on Fermi-Dirac Integrals*,Purdue University, June 27, 2008.

[22] http://cococubed.asu.edu/codepages/fermidirac.shtml

[23] *Quantum Transport* Nanoscience.

[24] T. Wang, Z. Qi, C. A. Moritz, *Opportunities and challenges in application-tuned circuits and architectures based on nanodevices*, in "First ACM International Conference On Computing Frontiers", pp. 503–511, aprile 2004.